REMARKS

Claims 1-9, 11-19, 21, 24-26, 29-30, and 34-36 were presented and examined. In response to the Office Action, Claims 1, 11, 13, 21, 24-26, 29-30, and 34 are amended, no claims are cancelled and no claims are added. Applicant respectfully requests reconsideration of pending Claims 1-9, 11-19, 21, 24-26, 29, 30, and 34-36 in view of at least the following remarks.

I. Claim Rejections Under 35 U.S.C. §112

Claims 21, 24-26, and 29-30 are rejected under 35 U.S.C. §112, first paragraph, as failing to comply with the written description requirement, specifically with regard to the term "program-thread-partition loop."

In the response, Claims 21 and 26 are amended to recite program-threads. In addition, the claims are amended to recite that program-threads may include program-thread loops. Support for such an amendment is provided at least with reference to FIG. 4 and paragraphs 0033-0035, which refer to program-threads 300, such as program-thread 300-1 (FIG. 3A) and program-thread 300-2 (FIG. 3B). We submit that based on such passages, one of skill in the art could easily read Applicant's specification and determine the various program-threads referred to by Claims 21 and 26, and that the loops within such program-threads may be referred to as program-thread loops, which may include identified critical sections, as recited by the pending claims.

Therefore, we submit that the terms "program-threads" and "program-thread loops" of Claims 21 and 26 recite a subject matter which was described in the specification in such a way as to reasonably convey, to one skilled in the relevant art, that the inventors at the time the application was filed had possession of the claimed invention. Therefore, please reconsider and withdraw the 35 U.S.C. §112 rejection of Claims 21 and 26 as well as of dependent claims 24-25 and 29-30.

V. Claim Rejections Under 35 U.S.C. §103

Claims 1-2, 11-12, 21, 24, 26, 29, and 34-35 are rejected under 35 U.S.C. §103(a) as being anticipated by Sohi et al, ("Multiscalar Processors", 1995, ACM 0-89791-698; pp. 414-425) ("Sohi"). Applicants respectfully traverse this rejection.

Claim 1 recites:

> 1. A method comprising:
> building a control flow graph (CFG) for a loop body of a sequential application program to form a CFG loop;
> updating nodes of the CFG loop to enclose identified critical sections of the sequential application program within corresponding pairs of boundary instructions; and
> modifying the updated nodes of the CFG loop reduce an amount of instructions between the corresponding pairs of boundary instructions to form a modified CFG loop;
> generating a plurality of application program-threads according to the modified CFG loop;
> concurrently executing the plurality of application program-threads that are generated from the modified CFG loop ; and
> synchronizing execution of the identified critical sections of the sequential application program among the plurality of concurrently executing application program-threads to ensure that the identified critical sections are executed in a sequential thread order. (Emphasis added.)

Sohi is generally directed to a multi-scalar execution model that partitions a control flow graph (CFG) into portions called tasks. Sohi describes that a program is statically poartitioned into tasks, which are defined as follows (see page 415, left column, third paragraph):

> A task is a portion of the CFG whose execution corresponds to a contiguous region of the dynamic instruction sequence (*e.g.*, part of a basic block, a basic block, multiple basic blocks, a single loop iteration, an entire loop, a function call, etc.).

In contrast with Claim 1, Sohi does not disclose or suggest modifying the updated nodes of the CFG loop to **reduce an amount of instruction between corresponding pairs of boundary instructions to form a modified CFG loop**, as in Claim 1. Sohi does disclose establishment of a large and accurate dynamic window of instructions from which independent instructions can be extracted and scheduled for parallel execution (see page 415, first paragraph under section 2.1); however, that is something different from modifying the updated nodes of the

CFG loop to reduce an amount of instructions between corresponding pairs of boundary instructions to form a modified CFG loop, as in Claim 1.

Moreover, Sohi cannot teach a reduction of the amount of instruction between corresponding pairs of boundary instructions as in Claim 1, since Sohi describes that the objective in multi-scalar execution is the assignment of a task to a processing unit and to proceed to the next task, without inspecting the contents of the assigned task. In addition, the description of the tag bits, as forward and stop bits (to mark the boundaries of a task), is simply a means to identify for a processing unit which dynamic instructions comprise its assigned task. Also, the partitioning referred to by the Examiner occurs after the boundaries and control edges of a task are known. (See page 14, right column, third paragraph.) We submit that the disclosure of Sohi is limited to the static partitioning of a program into tasks which are assigned to processing units for execution, without concern for the actual contents of the tasks. (See page 415, left column, second paragraph, last two sentences.) The static partitioning of an application program into specified sets of instructions, referred to as tasks that are assigned to processing units for parallel execution without regard for the contents of such tasks, cannot be properly interpreted as disclosing the modification of the updated nodes of a CFG loop to reduce the amount of instructions between corresponding pairs of boundary instructions to form a modified CFG loop, as in Claim 1, since such reduction would require further analysis of the task contents.

Moreoever, Sohi explicitly distinguishes itself from a multi-threaded processor, which according to Sohi is different in that the different threads executing on the multi-scalar processor are related as different parts of the sequential walk through the same program and are not control and data independent. (See page 422, right column, second paragraph.) Hence, the static partitioning of a program into tasks cannot be properly interpreted as generating a plurality of application program threads from the modified CFG loop, as in Claim 1.

In contrast with Claim 1, Sohi cannot disclose or suggest generating a plurality of application program-threads from the modified CFG loop, as in Claim1. (See Applicant's specification, pg. 8, para. 0043.) The Examiner has not identified and Applicants are unable to discern any disclosure, teaching, or suggestion regarding generating a plurality of application program-threads from the modified CFG loop, much less a reduction of the amount of instruction

between corresponding pairs of boundary instructions, as in Claim 1. We submit that neither sections 2.1, 2.2, nor any other portion of Sohi teaches or suggests generating a plurality of application program-threads from the modified CFG loop, much less a reduction of the amount of instruction between corresponding pairs of boundary instructions, as in Claim 1.

For each of the above reasons, Claim 1, and all claims which depend from Claim 1, are patentable over the cited art. Therefore, Applicant respectfully requests that the Examiner reconsider and withdraw the §103(a) rejection of Claims 1 and 2.

Each of Applicants' other independent claims includes features similar to those highlighted above in Claim 1. Therefore, all of Applicants' other independent claims, and all claims which depend on them, are also patentable over the cited art, for similar reasons. Consequently, we request that the Examiner reconsider and withdraw the §103(a) rejection of Claims 11-12, 21, 24, 26, 29, and 34-35.

Claims 3-9, 13-19, 25, 30, and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sohi in view of Vijaykumar T.N., ("Compiling for the Multiscalar Architecture", University of Wisconsin, 1998, pp. 1-191) "(Vijaykumar"). Applicants respectfully disagree with the Examiner's assertions and characterizations of the cited reference, and therefore traverse this rejection.

DEPENDENT CLAIMS

In view of the above remarks, a specific discussion of the dependent claims is considered to be unnecessary. Therefore, Applicant's silence regarding any dependent claim is not to be interpreted as agreement with, or acquiescence to, the rejection of such claim or as waiving any argument regarding that claim.

CONCLUSION

In view of the foregoing, it is submitted that Claims 1-9, 11-19, 21, 24-26, 29-30, and 34-36, as amended, patentably define the subject invention over the cited references of record, and are in condition for allowance and such action is earnestly solicited at the earliest possible date. If the Examiner believes a telephone conference would be useful in moving the case forward, he is encouraged to contact the undersigned at (310) 207-3800.

If necessary, the Commissioner is hereby authorized in this, concurrent and future replies, to charge payment or credit any overpayment to Deposit Account No. 02-2666 for any additional fees required under 37 C.F.R. §§1.16 or 1.17, particularly, extension of time fees.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR, & ZAFMAN LLP

Dated: October 23, 2008

By: _____
Joseph Lutz, Reg. No. 43,765

1279 Oakmead Parkway
Sunnyvale, California 94085-4040
Telephone (310) 207-3800
Facsimile (408) 720-8383

CERTIFICATE OF TRANSMISSION
I hereby certify that this correspondence is being submitted electronically via EFS Web on the date shown below.

_____
Alexandra Y. Caluen October 23, 2008